UPIC a Problem-Solving Framework: Understand, Plan, Implement, and Correctness/Debugging

Sadhana Suryadevara
sadhana.suryadevara@duke.edu
Duke University
Durham, North Carolina, USA

Kristin Stephens-Martinez
ksm@cs.duke.edu
Duke University
Durham, North Carolina, USA

CCS CONCEPTS

• **Social and professional topics** → *CS1*; **Computer science education**.

KEYWORDS

office hours, problem-solving process, cs1, data science

ACM Reference Format:
Sadhana Suryadevara and Kristin Stephens-Martinez. 2022. UPIC a Problem-Solving Framework: Understand, Plan, Implement, and Correctness/Debugging. In *Proceedings of the 2022 ACM Conference on International Computing Education Research V.2 (ICER 2022), August 7–11, 2022, Lugano and Virtual Event, Switzerland.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3501709.3544286

1 BACKGROUND AND CONTEXT

Some courses explicitly teach a problem-solving process [1, 2]. These processes provide scaffolding as students solve problems. However, to understand where students get stuck in a process, we must compare courses regardless of their problem-solving process.

2 OBJECTIVES

We present a framework to split the problem-solving process into four phases: (1) Understand the problem, (2) create a Plan, (3) Implement the plan, and (4) verify Correctness/debug (UPIC). We applied this framework to survey responses students provided before joining an online OH queue [3] for a CS1 and intermediate data science (DS) course. Our goal was to understand which phases students sought help.

3 METHOD

For each OH interaction, students reported their current UPIC phase in the pre-survey. For CS1, the question had one option per phase using the 7-steps terminology [1], a problem-solving process explicitly taught in that class. The DS course did not have a problem-solving process. So the teachers used UPIC to design options to replace an open textbox. See Table 1 for more details. For CS1, we have four semesters of data, Fall 2020 (Fa20) to Spring 2022 (Sp22), and three semesters for the DS, Spring 2021 (Sp21) to Sp22. We collected all at Duke University, a medium private R1 university.

Table 1: Pre-question options

Course	UPIC	Option
CS1	Understand	Doing an instance of the problem (Step 1 of the 7-steps)
	Plan	Developing a plan to solve a problem (Steps 3 and 4 of the 7-steps)
	Implement	Writing the code to solve a problem (Step 5 of the 7-steps)
	Correctness	Testing my program (Step 6 of the 7-steps)
DS	Understand	Understanding a problem or directions
	Understand	Understanding a concept from class
	Plan	Planning how to solve a problem before getting into the math/code details
	Implement	Writing the math/code details to solve a problem
	Correctness	Validating/testing/debugging my solution

4 FINDINGS

For three of the CS1 semesters, students sought the most help with implementation, see Figure 1. Correctness was usually the second most common, while understand and plan were the lowest. For DS, we found much greater variation, see Figure 2. For Sp21, understand was the most common phase, while for later semesters plan was significantly the lowest reported and the rest with no significant order. We suspect this is due to having an autograder introduced in Fa21.

5 IMPLICATIONS

The UPIC framework enables aggregating different problem-solving processes for an apples-to-apples comparison. We found that CS1 students usually seek the most help with implementation. For our DS course, understand was the most common phase when the course had no autograder. Once the course had an autograder, the phase proportions are more varied, suggesting that an autograder influences when in the problem-solving process students seek help.

Teachers that do not teach a problem-solving process could use UPIC as a framework to understand where students struggle or to create their own problem-solving process.

ACKNOWLEDGMENTS

We thank Brandon Fain and the support of the National Science Foundation grant #1934965.

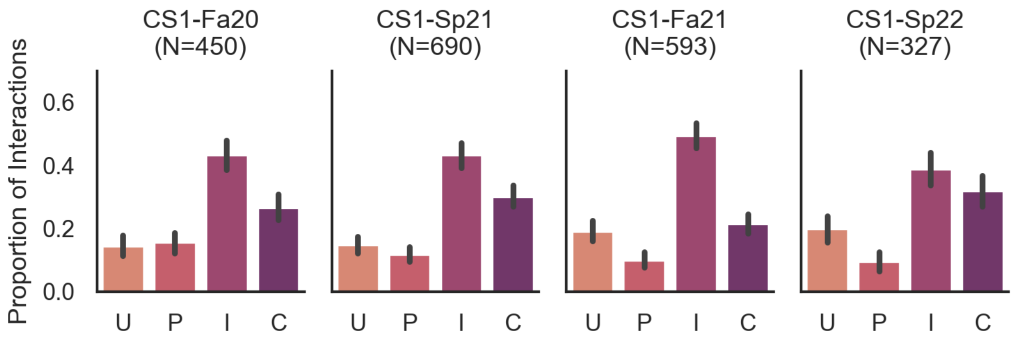

Figure 1: UPIC phase distribution for CS1 (N = # interactions).

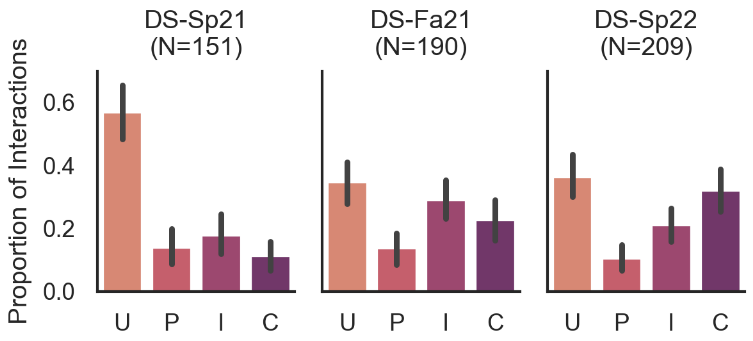

Figure 2: UPIC phase distribution for DS (N = # interactions).

REFERENCES

[1] Andrew D. Hilton, Genevieve M. Lipp, and Susan H. Rodger. 2019. Translation from Problem to Code in Seven Steps. In *Proceedings of the ACM Conference on Global Computing Education* (Chengdu,Sichuan, China) *(CompEd '19)*. Association for Computing Machinery, New York, NY, USA, 78–84. https://doi.org/10.1145/3300115.3309508

[2] Yanyan Ren, Shriram Krishnamurthi, and Kathi Fisler. 2019. What Help Do Students Seek in TA Office Hours?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 41–49. https://doi.org/10.1145/3291279.3339418

[3] Aaron J. Smith, Kristy Elizabeth Boyer, Jeffrey Forbes, Sarah Heckman, and Ketan Mayer-Patel. 2017. My Digital Hand: A Tool for Scaling Up One-to-One Peer Teaching in Support of Computer Science Learning. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 549–554. https://doi.org/10.1145/3017680.3017800